

Исследование трафика облачных приложений виртуальной реальности¹

Е.С. Корнеев, М.В. Любогощев, Е.М. Хоров

Институт проблем передачи информации им. А.А. Харкевича РАН, Москва

Поступила в редколлегию 15.03.2022 г.

Аннотация—Трафик облачных интерактивных приложений виртуальной реальности будет значительно отличаться от трафика, распространенного в современной сети Интернет, по структуре и требованиям к качеству обслуживания. Для совершенствования технологий передачи данных и разработки решений, нацеленных на обеспечение требуемого качества обслуживания трафика виртуальной реальности, необходимо разработать комплекс моделей, описывающий новый тип трафика и поведение пользователей. Данные модели могут быть использованы для оценки влияния различных показателей производительности сети (например, пропускной способности канала, задержки при передаче пакетов, надежности их доставки) на качество восприятия видеопотока виртуальной реальности конечным пользователем. В данной работе изучаются характеристики трафика распространенных приложений виртуальной реальности и разрабатывается подробная и гибкая модель приложения виртуальной реальности. С помощью данной модели можно генерировать видеопотоки виртуальной реальности, имеющие те же свойства, что и генерируемые существующими приложениями виртуальной реальности. Исследуется, как параметры кодирования видеопотока виртуальной реальности влияют на эффективность его доставки по беспроводной сети. Демонстрируется, что правильный выбор этих параметров позволяет существенно увеличить емкость сети по сравнению с применяемыми в существующих облачных приложениях виртуальной реальности.

КЛЮЧЕВЫЕ СЛОВА: Виртуальная реальность, интерактивные приложения, кодирование видеоизображения, емкость сети, имитационное моделирование.

DOI: 10.53921/18195822_2022_22_2_42

1. ВВЕДЕНИЕ

Технологии виртуальной реальности (англ.: VR) находят свое применение в различных областях: в сфере развлечений [1],[2], медицине [3],[4], инженерном деле [5],[6] и многих других. Во многих случаях, видеоизображение VR генерируется на сервере или в облаке, а затем доставляется на гарнитуры клиентских устройств. Полное погружение в виртуальную реальность возможно лишь при гарантировании высокого качества видеоизображения и низкого времени отклика на действия пользователя, и обеспечение надлежащего качества восприятия виртуальной реальности остается открытой проблемой. Разработка методов доставки трафика VR с требуемым качеством нуждается в достаточно адекватной и гибкой модели облачных приложений виртуальной реальности.

Трафик приложений виртуальной реальности включает в себя несколько потоков: поток команд и телеметрической информации, видеопоток (динамическое стереоизображение), аудиопоток. При этом именно доставка видеоизображения является наиболее ресурсоемкой из-за большого объема передаваемых данных и требуемой низкой задержки.

¹ Исследование выполнено в ИППИ РАН за счет гранта Российского научного фонда No 21-79-10431, <https://rscf.ru/project/21-79-10431/>.

Как правило, известные в литературе исследования обслуживания VR-трафика предполагают слишком упрощенную структуру генерируемого видеотрафика. К примеру, авторы работы [7] исследовали передачу VR-видеопотоков по USB кабелю между сервером и смартфоном. Специализированные высокопроизводительные шлемы виртуальной реальности значительно превосходят современные смартфоны по своим вычислительным мощностям, поэтому трафик, изучаемый в работе, отличается от генерируемого популярными облачными VR-приложениями.

В другой работе [8], исследующей обслуживание VR-видеопотоков, рассматривается трафик VR-приложений как последовательность случайных запросов, которые должны быть удовлетворены за время воспроизведения кадра $\tau = 1/FPS$ (англ.: Frames per Second, FPS). Предлагаемая модель VR-приложения чрезмерно упрощает видеопоток, поскольку предполагает последовательные видеокadres независимыми, а также не учитывает наличие буфера хранения кадров на VR-гарнитуре.

Авторами работы [9] была предпринята попытка исследовать трафик, генерируемый популярным приложением для шлема виртуальной реальности Oculus Quest 2. Им удалось перехватить сетевые пакеты, передаваемые от сервера к гарнитуре, однако видеокadres из полученного потока данных восстановлены не были. В связи с этим выводы авторов были основаны на анализе размеров пакетов и длительностей интервалов между ними, что не позволяет построить достаточно подробную модель работы VR-приложения, учитывающую обратную связь (а именно, как хорошо сеть обслуживает VR-трафик) при генерации новых порций данных.

В ряде работ исследуются алгоритмы адаптации качества передаваемого видеоизображения с учетом канальных условий. Например, в работе [10] предлагается адаптивный алгоритм выбора разрешения видеоизображения, позволяющий снизить нагрузку на сеть и при этом обеспечить наилучшее качество восприятия виртуальной реальности пользователями. При исследовании работы алгоритма авторы используют модель трафика, аналогичного передаваемому сервисом YouTube, который может отличаться от генерируемого современными приложениями VR. Использование реалистичной модели трафика позволит точнее моделировать алгоритмы, подобные представленному в описываемой работе.

С другой стороны, в работе [11] предлагается аналитическая модель приложения виртуальной реальности. С ее помощью авторы могут оценить эффективность передачи трафика при заданных параметрах сети и требованиях к надежности передачи. Используемая при этом модель трафика не отражает особенности кодирования видеоизображения современными VR-приложениями и требует уточнений.

В данной работе проанализирован трафик современных VR-приложений и в результате разработана модель VR-приложения, учитывающая структурные особенности VR-трафика, а также возможную обратную связь между обслуживанием VR-трафика и генерацией новых порций данных. Данная модель позволяет исследовать передачу по различным сетям видеопотоков виртуальной реальности, обладающих различными заданными характеристиками. Помимо этого в работе исследуется, как выбор параметров кодирования видеоизображения влияет на эффективность передачи видеопотока по сети с применением беспроводных технологий.

Дальнейшее изложение построено следующим образом. В разделе 2 приводится описание трафика приложения виртуальной реальности, исследуемого в работе. Раздел 3 посвящен описанию разработанной модели приложения. Численные результаты, полученные с помощью имитационного моделирования, анализируются в разделе 4. В разделе 5 резюмируются основные результаты работы.

Таблица 1. Параметры видеопотоков Pico Streaming Assistant

	72 Гц	90 Гц
Fast	8,5	8,0
Standard	43	40
HD	43	50

(а) Битрейт, Мбит/с

	72 Гц	90 Гц
Fast	5	5
Standard	5	5
HD	37	37

(б) Размер GoP

2. ТРАФИК ИНТЕРАКТИВНЫХ VR-ПРИЛОЖЕНИЙ

В данной работе изучается трафик, передаваемый одиночному пользователю во время игры в Steam VR с использованием беспроводной гарнитуры Pico Neo 2 [12]. Трафик перехватывался при помощи утилиты Wireshark [13].

Для формирования трафика применялось приложение Pico Streaming Assistant. Данное приложение позволяет выбирать качество передаваемого видеопотока и частоту обновления экрана из декартового произведения $\{\text{Fast, Standard, HD}\} \times \{72\text{Гц, } 90\text{Гц}\}$.

Приложение Pico Streaming Assistant передает два видеопотока одновременно, предназначенных для левого и правого глаза пользователя, соответственно. Помимо видеопотоков можно выделить периодический аудиопоток, а также несколько потоков в восходящем канале, соответствующих передаче данных о положении головы и рук пользователя и направлении его взгляда. Для передачи большей части данных между гарнитурой и сервером используется протокол RTP, незначительный объем данных передается от гарнитуры к серверу в UDP-потоках без использования RTP. Доминирующую нагрузку на сеть оказывает видеопотоки от сервера к гарнитуре, поэтому основное внимание в данной работе будет уделено именно данным потокам. Значения битрейтов видеопотоков приведены в Таблице 1(а).

Каждый из двух видеопотоков кодируется независимо с применением стандарта HEVC [14]. Данный стандарт позволяет кодировать кадр видеоизображения неперекрывающимися сегментами (англ.: Slice), каждый из которых может быть декодирован независимо или с опорой на декодированные ранее сегменты². Кадр, полностью состоящий из независимо декодируемых сегментов (англ.: Instantaneous Decoding Refresh, IDR), будем называть опорным, или IDR-кадром, остальные же кадры будем называть разностными. В передаваемом видеопотоке опорные кадры появляются строго периодически. Данная повторяющаяся последовательность называется GoP (англ.: Group Of Pictures), а количество образующих ее кадров — длиной, или размером GoP. Длина GoP отличается для различных настроек Pico Streaming Assistant и не может быть выбрана пользователем явно, и остается неизменной в процессе игры. Длины GoP, соответствующие различным режимам игры, указаны в таблице 1(б).

Видеопотоки, предназначенные различным глазам, должны быть синхронизированы, чтобы воспроизводимые изображения описывали виртуальную реальность, соответствующую одному и тому же моменту времени. Ограничений на согласование структур потоков при этом не накладывается: время между передачей двух соседних IDR-кадров, принадлежащих различным потокам, может быть любым в пределах от 0 до $t_{GoP} = GoP/FPS$.

В результате эксперимента было установлено, что текущая версия Pico Streaming Assistant кодирует видеоизображение единым сегментом. Необходимо также отметить, что адаптации видеоизображения под каналные условия во время игры не наблюдалось.

Функция распределения размеров передаваемых кадров представлена на рис. 1. Из того, что распределение имеет бимодальную структуру, можно сделать вывод, что видеопоток состоит из относительно небольших разностных кадров и опорных кадров, значительно превосходящих разностные кадры в размере. На графиках, соответствующих режимам игры Standard и

² Стандарт также определяет другие типы сегментов, однако при кодировании видеопотоков реального времени они не используются.

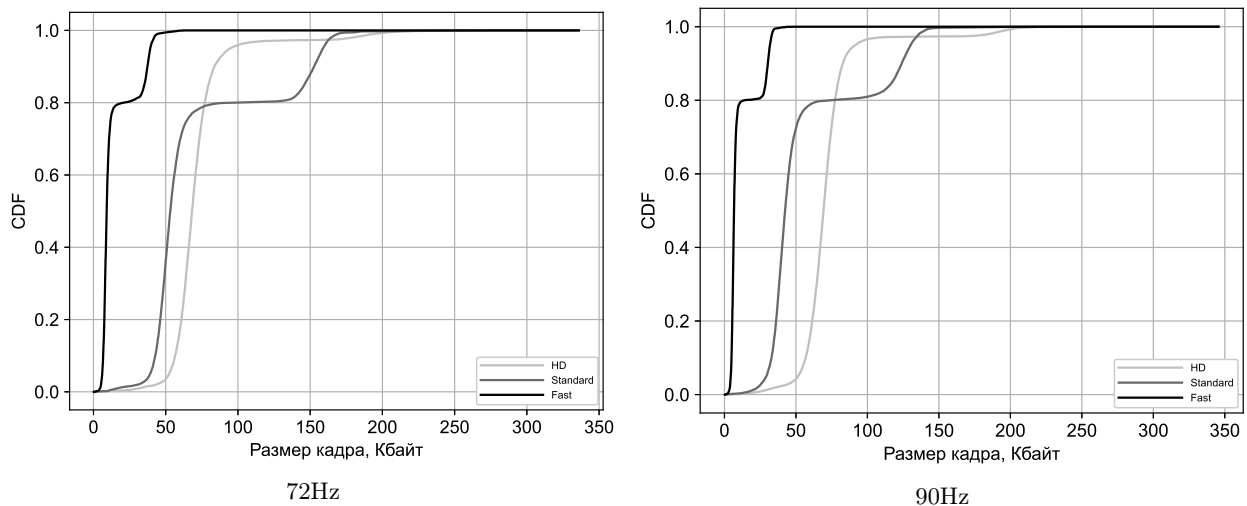


Рис. 1. Функция распределения размеров кадров Pico Streaming Assistant.

Fast, заметно, что вклад первой моды в функцию распределения равен 0.8, что согласуется с размером GoP = 5, т.е. тем, что на 4 разностных кадра приходится 1 IDR-кадр. Тот же эффект заметен и на графиках, описывающих режим HD, где вклад первой моды составляет $36/37 \approx 0.97$, что также соответствует размеру GoP = 37.

Приведенные выше рассуждения также подтверждаются рассмотрением временной развертки передаваемого видеопотока.

3. МОДЕЛЬ VR-СИСТЕМЫ

В данном разделе приводится описание разработанной имитационной модели облачного VR-приложения. При ее разработке за основу брались результаты, полученные в разделе 2. Модель разрабатывалась в рамках среды имитационного моделирования NS-3 [15].

Разработанная модель приложения состоит из двух частей, моделирующих взаимодействие клиентского приложения с сервером граничных вычислений. В начале VR-сессии клиент выбирает случайным образом видеодорожку, загружает доступные битрейты и запрашивает у сервера желаемое видеоизображение с одним из возможных битрейтов. Модель предусматривает возможность выбора распределений длительности сессии, ее продолжительности и перерыва между последовательными сессиями. Также возможно случайным образом выбирать начальное смещение в передаваемом видеопотоке, что позволяет моделировать передачу различных видеопотоков с заданными параметрами без необходимости перекодирования большого набора видеоклипов. В ответ на полученный запрос сервер начинает передавать кадры видеодорожки по очереди с заданной частотой $\nu = 1/FPS$. Как было указано в разделе 2, данные, соответствующие видеоизображению, обеспечивают основную нагрузку на сеть, поэтому на их фоне можно пренебречь остальным трафиком.

Важной особенностью модели приложения является возможность генерации видеопотока с заданными характеристиками. Разработанная модель использует видеокодек Nvidia NVENC [16] для получения требуемого видеоизображения. Подобрав параметры кодирования, можно генерировать видеопоток, точно совпадающий с описанным в разделе 2, или же обладающий другими требуемыми характеристиками. В частности, помимо частоты генерации кадров и желаемого битрейта видеопотока можно указать длину GoP, количество сегментов на кадр и максимальный размер кадра в байтах, определяемый глубиной VBV-буфера (англ.: Video Buffer Verifier, VBV). Эксперименты с приложением Pico Streaming Assistant показывают, что

Таблица 2. Параметры кодека

Параметр	PICO			Nvidia NVENC
	Fast	Standard	HD	
Длина GoP Intra Refresh	5 Нет	5 Нет	37 Нет	∞ Да
Битрейт (144Гц), Мбит/с	17	86	86	{10,20,40,60,80, 120,160,200}
Битрейт (180Гц), Мбит/с	16	80	100	{10,20,40,60,80, 120,160,200}
VBV	3-5 кадров			1 кадр (<i>bitrate/fps</i>)
Стандарт кодирования	HEVC			
Высота x Ширина, пикс.	1024x1080, битрейт < 40 Мбит/с 1832x1920, битрейт \in [40, 160] Мбит/с 3840x2160, битрейт > 160 Мбит/с			
Rate Control	CBR			
B-кадры	Нет			

при различных запусках приложения видеопотоки сдвинуты относительно друг друга на различное расстояние. Данное наблюдение позволяет считать видеопотоки независимыми и вместо передачи двух потоков моделировать передачу одного потока с частотой кадров, в два раза превышающей целевое значение. Это соответствует наиболее благоприятному сценарию, при котором опорные кадры разных видеопотоков передаются по очереди на равных расстояниях $t_{GoP}/2$ друг от друга. Обычно IDR-кадры значительно превосходят в размере разностные кадры, поэтому наибольшая их удаленность друг от друга приводит к наименьшей величине $p2a$ (англ.: Peak To Average) и, соответственно, более равномерному потоку данных.

Одним из известных способов уменьшения величины $p2a$ является применение технологии Intra Refresh. Технология предлагает отказаться от передачи полноценного опорного кадра в пользу составления видеопотока исключительно из разностных кадров, каждый из которых будет включать небольшую независимо декодируемую область изображения. Данный подход избавляет от периодических всплесков интенсивности передаваемого трафика за счет увеличения размера каждого из разностных кадров. Предлагаемые компанией Nvidia настройки видеокodeка для генерации видеопотока реального времени включают данную опцию.

В данной работе сравниваются два набора параметров кодирования видеоизображения:

1. аналогичные Pico Streaming Assistant;
2. Nvidia Low-latency [16].

Первый набор соответствует видеоизображению, передаваемому при использовании Pico Streaming Assistant. Второй набор соответствует настройкам кодека, рекомендованным разработчиками видеокodeка — компанией Nvidia, для кодирования видеоизображения, предназначенного для трансляции с малыми задержками. Параметры кодека, используемые в обоих случаях, представлены в таблице 2.

Каждое видеоизображение кодируется с применением нескольких наборов параметров видеокodeка. Видеопоток с заданными параметрами назовем представлением видеоизображения (англ.: Representation). Представление можно описать следующими параметрами:

1. FPS;
2. средний битрейт видеопотока;
3. максимальный размер кадра;
4. длительность видеопотока в секундах;

5. высота изображения в пикселях;
6. ширина изображения в пикселях.

Полученный в результате кодирования видеопоток сохраняется в базе данных в виде последовательности сегментов кадров. Для моделирования передачи кадров используется следующая информация о каждом сегменте:

1. размер в байтах;
2. порядковый номер кадра, которому принадлежит данный сегмент;
3. порядковый номер сегмента в текущем кадре.

Данной информации достаточно, чтобы приложение могло моделировать передачу видеопотока различным образом, включая независимую поочередную передачу каждого сегмента и передачу кадра как единого целого. Серверное приложение отвечает за разбиение передаваемых сегментов на RTP-пакеты с учетом ограничения на максимальный размер IP-пакета в сети и добавление RTP-заголовков, описанных в стандарте [17]. Клиентское приложение, в свою очередь, собирает каждый сегмент, разбитый на RTP-пакеты, моделирует воспроизведение полученных видеок кадров и осуществляет адаптацию битрейта на основе собираемых статистик. Если видеок кадр был получен не полностью или после момента, когда он должен был быть воспроизведен, приложение считает этот кадр потерянным.

4. ЧИСЛЕННЫЕ РЕЗУЛЬТАТЫ

Разработанная модель была реализована в среде имитационного моделирования NS-3 [15] и применена для оценки качества методов кодирования видеоизображения, предложенных в разделе 3. В данной работе рассматривались два независимых эксперимента. Эксперимент, описанный в разделе 4.1, использовался для исследования влияния выбранного способа кодирования видеоизображения на эффективный битрейт видеопотока, то есть на минимальную пропускную способность сети, достаточную для доставки видеопотока без потерь. В эксперименте, описанном в разделе 4.2, была продемонстрирована эффективность предложенного метода кодирования при наличии множества VR-потоков в сети.

4.1. Сценарий с проводным соединением

Рассмотрим передачу видеопотока между сервером и единственным клиентом, соединенными надежной проводной связью, обладающей постоянной пропускной способностью. Будем считать, что потеря данных при передаче не происходит, поэтому все потери связаны с превышением допустимого времени на доставку кадра, составляющим 20 мс [18]. Клиент выбирает частоту кадров и битрейт и запрашивает у сервера соответствующий видеопоток. VR-сессии инициируются клиентом в случайные моменты времени, продолжительность сессии выбирается случайным образом из равномерного распределения и составляет от 90 до 110 секунд. По завершении сессии пользователь ждет время, случайным образом выбранное из усеченного экспоненциального распределения со средним значением 30с, минимальным и максимальным значениями 10 и 60 с, соответственно. Подобное поведение пользователей характерно в случае многопользовательских игр, одновременной работы нескольких инженеров в одном помещении и в иных ситуациях. Длительность эксперимента составляет 1000с, каждый набор параметров повторяется 10 раз.

На рис. 2 представлены зависимости доли потерянных кадров от пропускной способности соединения. Из рис. 2 (а) видно, что для передачи видеоизображения, закодированного способом, применяемым в приложении Pico Streaming Assistant, необходима пропускная способность, значительно превышающая средний битрейт видеопотока, причем отличие тем больше,

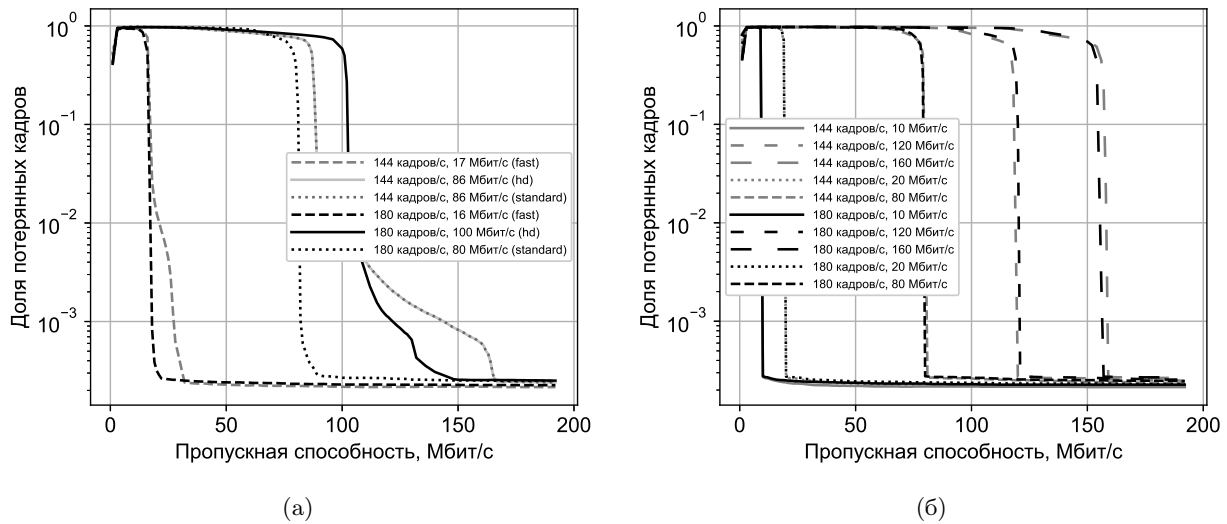


Рис. 2. Влияние пропускной способности соединения на потери VR-трафика: (а) – Pico, (б) – Nvidia NVENC

чем выше битрейт передаваемого видеоизображения. Например, для обеспечения доли потерь кадров, не превышающей 10^{-3} в наилучшем доступном разрешении 100 Мбит/с необходима пропускная способность соединения от 150 Мбит/с. Аналогично, за счет присутствия в видеопотоке редких, но больших IDR-кадров передача видеопотока в стандартном разрешении с частотой 144 кадра/с требует пропускную способность от 170 Мбит/с, что в два раза превосходит целевой битрейт видеопотока. В то же время из рис. 2 (б) видно, что для передачи видеопотоков Nvidia NVENC достаточна пропускная способность канала, точно соответствующая битрейту передаваемого видеоизображения во всем диапазоне исследуемых битрейтов. Кодирование видеоизображения с применением настроек, предлагаемых Nvidia, обеспечивает отсутствие в потоке больших кадров, которые необходимо доставить за отведенные 20 мс, и в результате меньше сетевых ресурсов потребляется при передаче. В случае многопользовательского приложения, разделяющего общий беспроводной канал, уменьшение объема потребляемых сетевых ресурсов позволяет одновременно обслуживать большее число пользователей, или же передавать более качественный видеопоток небольшому числу пользователей. Таким образом, за счет применения настроек видеокodeка, предлагаемых Nvidia, можно эффективно увеличить емкость сети для VR-потоков без внесения значительных изменений в логику работы приложения.

4.2. Многопользовательский сценарий

Рассмотрим теперь сценарий, в котором несколько малоподвижных пользователей одновременно используют приложение виртуальной реальности, будучи подключенными к одной беспроводной точке доступа. Такое расположение устройств может соответствовать, к примеру, одновременной работе инженеров на предприятии.

В рамках эксперимента пользовательские устройства случайным образом располагаются в прямоугольной комнате со сторонами 24 и 12 метров, в середине которой расположена точка доступа Wi-Fi 6. Каждый пользователь прямолинейно двигается в пределах квадрата со стороной 1м, очерченного вокруг начального положения. При достижении границы зоны пользователь продолжает движение в случайном направлении внутри нее с постоянной скоростью 5 м/с. Точка доступа использует SU-MIMO для передачи данных пользовательским устройствам.

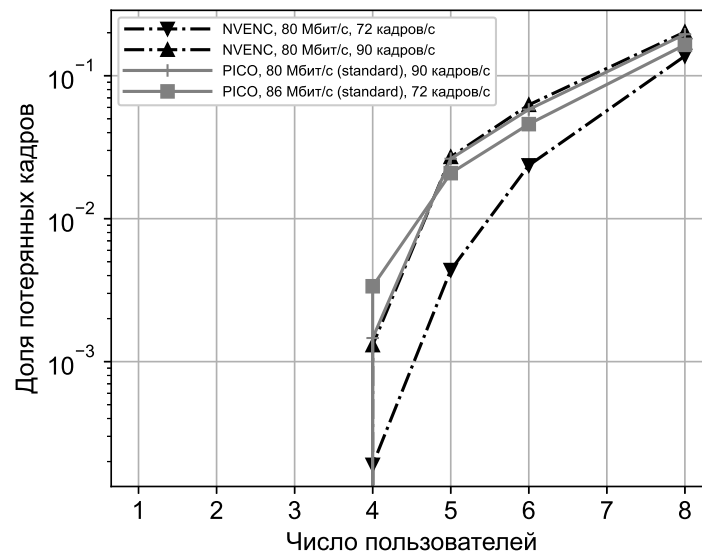


Рис. 3. Доля потерянных видеокладов

Параметры VR-сессий соответствуют описанным в разделе 4.1. Ограничение на время доставки кадра уменьшено до 15 мс, что соответствует наличию буфера на пользовательском устройстве, вмещающего один кадр.

Результаты численного моделирования передачи видеопотоков приведены на рис. 3. Указанные значения битрейтов являются целевыми для видеокодека, в действительности средний битрейт полученного видеопотока может отличаться от ожидаемого. Для сравнения способов кодирования в данном эксперименте были выбраны представления видеоизображения, обладающие одинаковым фактическим битрейтом, то есть Pico Standard и NVENC, 80Мбит/с для частот 72 и 90 кадров/с.

Из сравнения зависимости доли потерянных кадров от числа пользователей в сети, приведенных на рис. 3 для частоты 72 кадра/с, следует, что, при достаточно малом числе пользователей, емкости сети достаточно для успешной передачи всех пакетов. С увеличением количества пользователей доля потерянных кадров растет, при этом видеопоток, полученный с применением настроек, предлагаемых Nvidia, теряет значительно меньше кадров по сравнению с аналогичным потоком, использующим настройки Pico Streaming Assistant. На частоте 90 кадров/с оба способа кодирования демонстрируют одинаковые результаты.

5. ЗАКЛЮЧЕНИЕ

В данной работе исследован трафик популярных приложений виртуальной реальности и разработана подробная и гибкая имитационная модель VR-приложения, позволяющая генерировать видеопотоки VR, имеющие те же свойства, что и генерируемые существующими приложениями виртуальной реальности. Проведено сравнение методов кодирования видеоизображения, в том числе рассмотрен способ кодирования, позволяющий эффективнее передавать видеопоток по сети с применением беспроводных технологий. При помощи разработанной модели продемонстрировано, что использование настроек видеокодека, предлагаемых компанией Nvidia, позволяет значительно снизить требования к качеству обслуживания VR потоков и увеличить емкость сети по сравнению с применяемыми в актуальных VR-приложениях.

СПИСОК ЛИТЕРАТУРЫ

1. Evaluating display fidelity and interaction fidelity in a virtual reality game / Ryan P McMahan, Doug A Bowman, David J Zielinski, Rachael B Brady // IEEE transactions on visualization and computer graphics. — 2012. — Vol. 18, no. 4. — P. 626–633.
2. Viewport-adaptive encoding and streaming of 360-degree video for virtual reality applications / Kashyap Kammachi Sreedhar, Alireza Aminlou, Miska M Hannuksela, Moncef Gabbouj // 2016 IEEE International Symposium on Multimedia (ISM) / IEEE. — 2016. — P. 583–586.
3. A systematic review of virtual reality simulators for robot-assisted surgery / Andrea Moglia, Vincenzo Ferrari, Luca Morelli et al. // European urology. — 2016. — Vol. 69, no. 6. — P. 1065–1080.
4. Augmented and virtual reality in dental medicine: A systematic review / T Joda, GO Gallucci, Daniel Wismeijer, NU Zitzmann // Computers in biology and medicine. — 2019. — Vol. 108. — P. 93–100.
5. Su Ping, Wang Shuo. Virtual reality practice in architecture design // 2012 IEEE Symposium on Electrical & Electronics Engineering (EESYM) / IEEE. — 2012. — P. 98–101.
6. Wolfartsberger Josef. Analyzing the potential of Virtual Reality for engineering design review // Automation in Construction. — 2019. — Vol. 104. — P. 27 – 37. — URL: <http://www.sciencedirect.com/science/article/pii/S0926580518312093>.
7. Lecci Mattia, Zanella Andrea, Zorzi Michele. An ns-3 implementation of a bursty traffic framework for virtual reality sources // arXiv preprint arXiv:2103.04609. — 2021.
8. Zhang Shulai, Tao Meixia, Chen Zhiyong. Exploiting caching and prediction to promote user experience for a real-time wireless VR service // 2019 IEEE Global Communications Conference (GLOBECOM) / IEEE. — 2019. — P. 1–6.
9. Virtual Reality Gaming on the Cloud: A Reality Check / Sihao Zhao, Hatem Abou-zeid, Ramy Atawia et al. // arXiv preprint arXiv:2109.10114. — 2021.
10. Liubogoshchev Mikhail, Korneev Evgeny, Khorov Evgeny. EVerEst: Bitrate Adaptation for Cloud VR // Electronics. — 2021. — Vol. 10, no. 6. — P. 678.
11. Adaptive Cloud-Based Extended Reality: Modeling and Optimization / Mikhail Liubogoshchev, Kamila Ragimova, Andrey Lyakhov et al. // IEEE Access. — 2021. — Vol. 9. — P. 35287–35299.
12. Pico Neo 2 HMD. — URL: <https://www.pico-interactive.com/us/neo2.html>.
13. Wireshark network protocol analyzer. — URL: <https://www.wireshark.org/>.
14. Series H: Audiovisual and multimedia systems: Infrastructure of audiovisual services—coding of moving video. High efficiency video coding : Rep. / ISO ; Executor: ITU-T Study Group et al.
15. The ns-3 network simulator. — URL: <http://www.nsnam.org/>.
16. NVIDIA Video Codec SDK. — URL: <https://developer.nvidia.com/nvidia-video-codec-sdk>.
17. Wenger S, Hannuksela MM. RTP payload format for high efficiency video coding (HEVC). — 2016.
18. Towards efficient edge cloud augmentation for virtual reality mmogs / Wuyang Zhang, Jiachen Chen, Yanyong Zhang, Dipankar Raychaudhuri // Proceedings of the Second ACM/IEEE Symposium on Edge Computing. — 2017. — P. 1–14.

A Study of Virtual Reality Traffic Interactive Applications

E.S. Korneev, M.V. Liubogoshchev, and E.M. Khorov

The traffic of cloud-based interactive virtual reality (VR) applications will differ significantly from the traffic typical for the modern Internet in terms of structure and quality of service requirements. To improve data transmission technologies and develop solutions aimed at providing the required quality of service for VR traffic, it is necessary to develop a set of models that describe a new type of traffic and user behavior. These models can be used to assess the impact of various network performance indicators (for example, channel bandwidth, packet transmission delays, reliability of their delivery) on the quality of experience of the VR video stream by the end user. This paper studies the traffic properties of common VR applications and develops a detailed and flexible VR application model. The model can be used to generate VR video streams that have the same properties as those generated by existing VR applications. The paper studies how the encoding parameters of a VR video stream affect the efficiency of its delivery over a wireless network. The correct choice of these parameters is demonstrated to significantly increase the network capacity compared with those used in existing cloud-based VR applications.

KEYWORDS: Virtual reality, interactive applications, video encoding, network capacity, simulation.